NASA Technical Memorandum 100084

Transport Delay Compensation for Computer-Generated Imagery Systems

Richard E. McFarland, Ames Research Center, Moffett Field, California

January 1988

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

Because of their computational overhead, CGI systems produce pure transport delay. The delay occurs downstream from the command environment of a typical mainframe computer within which vehicle states are computed. Without loss of generality, it may be assumed that these states are both numerically correct and properly synchronized with real time upon transmittal to the CGI system.

A given CGI system's pure transport delay is known or, as will be discussed, is measurable. It constitutes the time interval between the correctly synchronized command signals and their consequential visual presentation to the pilot.

Scene presentation delay is not very large for modern CGI equipment (on the order of 100 msec). However, it is perceived by a pilot as a summation with other (real aircraft) lags inherent in the total system. The military specification for piloted vehicle flying qualities (MIL-F-8785C), for instance, places at 100 msec the maximum allowable delay between cockpit control input and aircraft response for satisfactory Level 1 handling qualities. Clearly, a faithful mathematical model of a vehicle that just meets this requirement cannot be used in a simulation facility for Level 1 research if the CGI system itself has a 100 msec delay. Indeed, the additional CGI delay would move the simulation model into the region of Level 3 handling qualities.

It is usually accepted (Ref. 3) that ". . . if delays in ground-based engineering simulators used for evaluating the handling qualities of advanced aircraft cannot be appropriately compensated, the results will be misleading at best." As perceived by a pilot, CGI transport delay accumulates with other simulation delays (e.g., analog-to-digital conversion), as well as with real, faithfully simulated aircraft phase lags at any given operational frequency. Some aircraft are more responsive than others, and depending upon augmentation, their controllability may vary over a wide range. For instance, in studying control-system failures, the cumulative quantity of CGI delay may well constitute "the straw that breaks the camel's back." As opposed to lags caused by electrical, mechanical, and aerodynamic sources, this delay is a simulation fiction caused by the computational requirements of the CGI computer. As an additional lag component, it can degrade the simulation model so that it does not behave like the aircraft. CGI delays thus degrade a pilot's control capability during high-gain tasks.

Compensation Basis

It is hypothesized that in developing a visual compensation algorithm, the bandwidth of consideration need not be large. The essential postulation is that a pilot cannot manually control an aircraft based upon a visual presentation of terrain/objects beyond the frequency of a very few Hertz. An effective visual presentation filter is inherent in the fact that visual display activity is proportional to the double integral of pilot command activity. A "visual step response," for instance, need not

be considered (control steps do not produce visual steps). The inertial and aerodynamic characteristics of aircraft also contribute to the attenuation.

Hence, the hypothesis is formed: In the man-in-the-loop simulation environment, high-gain tasks are limited in frequency to perhaps two or three Hertz. Beyond this bandwidth, both human control capacity and correlated visual activity vanish. This hypothesis is the basis of the Ames Compensation Algorithm.

Under conditions where the hypothesis holds, a CGI compensation algorithm does not require broad-band applicability. If the system gain is correct and the phase is proper in an appropriate band, then performance beyond that band is immaterial because other system components cannot respond.

Two minor exceptions to the hypothesis have been identified in utilizing the compensation algorithm in general simulation. These exceptions are easily resolved, as will be shown.

The Ames Compensation Algorithm has a discrete basis. This basis concomitantly utilizes the specific velocity-to-position integration scheme used in the Flight Systems and Simulation Research Division at NASA Ames (for two decades), although similar schemes may be developed using other discrete integration assumptions or higher-order formulations. Continuous integration was assumed in the foundation material of Ref. 1; however, the resultant algorithmic coefficients are similar to those presented here. Also, the performance using these two fomulations is similar.

Continuous System Without Delay

This discussion considers transfer functions of the type that, in "perfect form," reduce to the continuous integration operation,

$$f(s)\Big|_{s=j\omega} = \frac{u(s)}{v(s)}\Big|_{s=j\omega} = \frac{1}{s}\Big|_{s=j\omega} = -j/\omega \qquad (1)$$

or "velocity-to-position transfer function." Here "u" is a position and "v" is a velocity, where "velocity" and "position" refer to both linear and angular measures. Transfer functions other than the perfect system are required because of transport delay, and because of the characteristics of discrete realizations. These functions exhibit "relative phase errors" (with respect to perfect integration) defined as the arctangent of the negative ratio of the real and imaginary components of equation (1):

$$\phi = \tan^{-1}\left[-\frac{Re(f)}{Im(f)}\right] \qquad (2)$$

By definition, the perfect system has zero phase error (the real part of equation (1) is zero).

Discrete, real-time aircraft simulation models are addressed here that have a constant cycle time "T," typically on the order of 20 or 30 msec. The transport delay interval "P" of a CGI sytstem is generally on the order of 100 msec. The following definitions are useful,

$$\theta = \omega T$$
$$\psi = \omega P \tag{3}$$

where ω is an arbitrary frequency limited only by the Nyquist criterion, θ is a "discrete frequency equivalent related to the cycle time of the mainframe computer," and ψ is a "discrete frequency equivalent related to the CGI (pipeline) transport delay."

Continuous System With Delay

The uncompensated velocity-to-position transfer function of equation (1) with the addition of a pure transport delay "P" may be expressed in Laplace notation by

$$f_U(s)\Big|_{s=j\omega} = \frac{e^{-sP}}{s}\Big|_{s=j\omega} = -\frac{\sin(\psi) + j\cos(\psi)}{\omega} \tag{4}$$

such that the relative phase error from equation (2) is:

$$\phi_U = \tan^{-1}\left[-\frac{\sin(\psi)}{\cos(\psi)}\right] = -\psi \tag{5}$$

This phase error is the reason for which compensation schemes are created.

Ames Integration With Delay

The prediction operation in simulation may be represented in terms of the integration operation. Figure 1 depicts the state integration scheme used at Ames, as well as other features to be discussed. The compensation filter is shown in Fig. 1, where z-transform notation is used because of its discrete basis. This filter augments the discrete velocity-to-position integration, and for this reason, is itself a function of the specific integration algorithm utilized. The velocity-to-position integration algorithm at Ames' simulation laboratory is therefore discussed.

In discrete models the integration operation must be performed by an algorithm, and algorithms introduce their own errors. As shown in Fig. 1, the velocity-to-position integration at Ames is handled by the trapezoidal integration algorithm. The appropriate transfer function with a superimposed transport delay "P" is:

$$f_E(z)\Big|_{z=e^{j\omega T}} = \frac{Tz^{-P/T}}{2}\left(\frac{1 + z^{-1}}{1 - z^{-1}}\right)_{z=e^{j\omega T}}$$

$$= - \frac{T/2}{\tan(\theta/2)} [\sin(\psi) + j \cos(\psi)] \qquad (6)$$

This corresponds to an uncompensated visual scene presentation in Fig. 1.

The trapezoidal algorithm does not of itself introduce any phase error. By comparison with equation (5) the relative phase error in this system (note the use of equation 2) is shown to be identical to that of the transport delay:

$$\phi_E = \phi_U = -\psi \qquad (7)$$

The absence of a phase error using the trapezoidal integration algorithm was the main reason for its selection at Ames for general aircraft simulation. This is an important consideration because phase differences between the position and velocity terms of aerodynamic buildup equations produce improper responses in simulation models. Incorrectly coded difference equations never converge to the correct differential equations, despite the size of T.

The trapezoidal algorithm does, however, introduce magnitude attenuation

$$|f_E| = \frac{T/2}{\tan(\theta/2)} \qquad (8)$$

but for reasonable cycle times (small T), this attenuation is not very large.

THE COMPENSATION ALGORITHM

Any number of previous values in a sequence may be utilized in developing a discrete prediction model, and various boundary conditions may be imposed to shape responses and determine the requisite parameters. Higher-order forumulations than that developed here have been investigated, but they distort responses too much in the region of interest. Lower-order formulations fail to meet the design objectives.

The Ames algorithm uses three successive values of velocity in its projection of position "P" seconds into the future. Perfect answers are

delivered for both zero frequency (constant velocity), and at a "crossover frequency (F_o)," some distance into the discrete bandwidth. The interval from zero to this frequency is the "pass band," over which the algorithm displays some good characteristics.

Algorithm Development

In the development of the compensation algorithm a three parameter extrapolation form given by

$$u_{k+P/T} = u_k + b_o v_k + b_1 v_{k-1} + b_2 v_{k-2} \tag{9}$$

is assumed, where this difference equation represents the predicted value of $z^{P/T}u(z)$ in Fig. 1. Equation (9) is tuned for the constant velocity condition by imposing the constraint:

$$P = b_o + b_1 + b_2 \tag{10}$$

This particular derivation differs from the foundation material of Ref. 1 at this juncture. Using trapezoidal integration, the transfer function representing equation (9) is written in z-transform notation:

$$f_A(z) = \frac{u(z)}{v(z)} = z^{-P/T}\left[\frac{T}{2}\left(\frac{1 + z^{-1}}{1 - z^{-1}}\right) + b_o + b_1 z^{-1} + b_2 z^{-2}\right] \tag{11}$$

All components of this function are shown in Fig. 1 in the transition from vehicle velocity to visual display.

The total delay-plus-prediction process may be compared to the perfect velocity-to-position transfer function by forming the relative error function,

$$h(\omega) = j\omega f_A(z)\Big|_{z=e^{j\omega T}} \tag{12}$$

which may be used to establish an additional constraint. This constraint is that equation (12) is "tuned" (i.e., it is error free at a particular frequency). The particular frequency is defined as the "crossover frequency" $F_o = \omega_o/2\pi$. As will be discussed, the crossover frequency is selected (not usual definition) to be as large as possible, considering errors within the interior band. At this frequency, two important parameters are defined:

$$\begin{aligned} \theta_o &= \omega_o T \quad \text{(the cyclic angle)} \\ \psi_o &= \omega_o P \quad \text{(the projection angle)} \end{aligned} \tag{13}$$

These angles are the "essential primitives" of the discrete system. For convenience in performing the required conversions of equation (13), Fig. 2 presents the cyclic angle θ_o versus cycle time "T" for various

values of the crossover frequency F_o. The interval from zero to F_o is referred to as the "pass band." The projection angle ψ_o in equation (13) may be obtained by multiplying the cyclic angle by the ratio $r = P/T$.

The value of equation (12) at the crossover frequency $F = F_o$ is a complex number. In combination with the constraint of equation (10), the second and third equations necessary to determine the three unknown parameters b_k are created by setting the phase of $h(\omega_o)$ equal to zero, and the magnitude of $h(\omega_o)$ equal to unity. Utilizing the impressed constraints, simultaneous solution then produces the coefficients

$$b_o = \frac{\{[\psi_o + \sin\psi_o(1 - 2\cos\theta_o)]\sin\theta_0 + [(1/2)\theta_o \sin\theta_o - \cos\psi_o(1 - \cos\theta_o)](1 + 2\cos\theta_o)\}}{2\omega_o \sin\theta_o(1 - \cos\theta_o)}$$

$$b_1 = \frac{\sin\theta_o[2\sin(\theta_o + \psi_o) - 2\psi_o\cos\theta_o - \theta_o(1 + \cos\theta_o)]}{2\omega_o \sin\theta_o(1 - \cos\theta_o)} \qquad (14)$$

$$b_2 = \frac{\sin\theta_o[\psi_o - \sin\psi_o + (1/2)\theta_o] - \cos\psi_o(1 - \cos\theta_o)}{2\omega_o \sin\theta_o(1 - \cos\theta_o)}$$

for use in equation (9) for simulation models (see Fig. 1). Compensated signals are sent to the CGI system only, and do not appear elsewhere in the model.

The algorithm's coefficients b_k are similar to equations (31) through (33) of Ref. 1. They differ only in that perfect integration (not trapezoidal) was assumed in the reference. "Velocity prediction," an option also mentioned in the reference, creates too much high-frequency noise, and is not used in the Ames simulation system. The superfluous concept of "velocity prediction" was intended to provide updated velocities to the CGI computer for later use in its "smoothing operation," as discussed below.

Architectural Considerations

A compensating (or predicting) algorithm may be utilized exclusively within the mainframe computer, within which the pertinent command position values are computed and properly synchronized with real time upon transmittal to the CGI system.

An alternate structure consists of delivering velocities as well as positions to CGI computational equipment, within which the required

compensation is then performed. This may or may not be the preferred architecture depending on storage, computational, or interface limitations associated with the CGI equipment.

The above configurations only account for a constant value of pure transport delay. An additional "smoothing operation" must be performed in the general case of asynchronous operations. The smoothing operation (discussed in Ref. 1) takes place exclusively in the CGI computers. The purpose of this operation is to synchronize (sort and linearly extrapolate) asynchronous signals between the mainframe and CGI computer system. In comparison with the large transport delay interval, the smoothing operation extrapolates signals over small (and variable) intervals. However, experience at Ames Research Center indicates that this operation should not utilize velocity values, and that if it does, the velocity values should definitely not be "predicted." Rather, it is better (within the CGI computer) to use the compensated position values for the smoothing operation. This has the added benefit of unburdening the interface.

Although the mainframe/CGI computer operations are asynchronous, both compensation and smoothing could be performed in one step within the CGI computer itself. Since the algorithm would reside in the CGI computer, this would require multiple buffers with appropriate "time stamps," as well as considerable CGI computational capacity. Using this architecture, the smoothing operation would be implicit. Because of variable "P," the coefficients of equation (14) would be nonstationary. This system deserves more study.

Phase Relationship

The compensation algorithm's performance may be determined using the previously defined "discrete frequency equivalents" θ and ψ. For instance, normalized coefficients c_k may be created

$$c_k = \omega_o b_k \quad (k = 0,\ 1,\ 2) \tag{15}$$

which are not explicit functions of T, P or ω_o. Using these coefficients the following expression for the phase error of the transfer function defined by equation (11) is developed:

$$\phi_A = \tan^{-1}\left\{\frac{\begin{array}{l}\sin\theta[c_o\cos\psi + c_1\cos(\theta+\psi) + c_2\cos(2\theta+\psi)] \\ \quad -(1/2)\theta_o\sin\psi(1+\cos\theta)\end{array}}{\begin{array}{l}\sin\theta[c_o\sin\psi + c_1\sin(\theta+\psi) + c_2\sin(2\theta+\psi)] \\ \quad +(1/2)\theta_o\cos\psi(1+\cos\theta)\end{array}}\right\} \tag{16}$$

An Example

For illustrative purposes consider the following example. Selecting a mainframe cycle time T = 20 msec and a crossover frequency F_o = 3 Hz, a cyclic angle of 21.6° is produced from either equation (13) or Fig. 2.

With a given CGI delay interval $P = 83$ msec, the algorithm's performance is as shown in Fig. 3.

The parameters used in creating Fig. 3 are those of a "DIG" (digital image generator) CGI system in use at Ames, and constitute a realistic case. Figure 3(a) displays the improvement in phase over the uncompensated system, i.e., equation (7) versus equation (16). Figure 3(b) displays the trade off in that it indicates how pass-band performance is retained using the algorithm by sacrificing performance beyond the pass band.

It is fundamental in the Ames approach that position commands have relatively low frequency content; hence, amplification beyond the pass band is relatively inconsequential. Although specific transfer functions are not used in this paper, in Ref. 1 a roll axis transfer function was used from the XV-15 Tilt Rotor that illustrates this phenomenon.

As an illustration of response limitations (Ref. 4), consider that researchers have been concerned about bandwidth reductions from 2.5 to 1.5 rad/sec, not Hertz. (In this case, the reduction was caused using a cycle time of 70 msec where the visual delay was 100 msec.) Degradation at these low frequencies was deemed "unacceptable for high-workload pilot tasking." These concerns now appear to have little foundation: the compensation algorithm operates over a much wider bandwidth. Practical human factors limitations are illustrated by statements such as "the pilot had trouble generating inputs at frequencies above 2 Hz" (Ref. 5).

Extrema

As shown in Fig. 3(a), a local algebraic minimum typically occurs in the phase error ϕ_m within the pass band ($F < F_o$), and an algebraic maximum occurs beyond the pass band. Note that within the pass band, the algebraic minimum corresponds to the maximum error.

For practical reasons, it may be required that the extreme pass-band phase error ϕ_m be controlled. For a given cycle time T and CGI delay P, it is possible to specify ϕ_m and determine F_o, or vice versa.

The phase error extremum in the pass band may be found by taking the derivative of ϕ_A in equation (16) with respect to ω and setting the result equal to zero. This produces a cubic transcendental equation in θ, which may be solved numerically. Any technique that brackets the root (such as the Bisection method or Regula falsi) easily isolates the pass-band root. Having obtained θ where ϕ_A is minimum, the extreme phase error in the pass band is itself obtained by substitution into equation (16).

Solutions to this process are presented in Fig. 4 for a wide variety of conditions. Relatively small extreme phase errors are generated for rather large projection angles, providing the cycle time is reasonable.

Since the projection angle $\psi_o = r\theta_o$, where the ratio $r = P/T$, Figs. 2 and 4 may be used together to determine either ϕ_m or F_o.

EXCEPTIONS

High frequencies in a simulation model can cause some blurring of the visual image, and this blurring may be distracting to the pilot under certain conditions. Two such cases have been identified during simulation studies at Ames. They do not involve steady state, distinct high frequencies such as the N/rev phenomenon of rotorcraft; rather, they are associated with intermediate frequencies. This is due to the relatively low pass band of the simulation model/CGI system.

Touchdown Transients

For a brief interval during touchdown, some blurring of the visual presentation may be noticed. This is caused by the broad band of frequencies generated by the landing shock.

The phenomenon of landing is difficult to model in digital simulations because stiff differential equations are activated with initial conditions that may be quite unrealistic. (This is a statement about probability, as discussed in Ref. 6, and is related to sample data theory.) Even without aggravated transients caused by sample data phenomena, for a short interval of time severe transients occur that span a wide band of frequencies.

The solution to this problem is quite simple. For a brief interval, perhaps three or four computer cycles, the compensation algorithm may be simply disengaged upon recognition of "weight on wheels." As an extension to this technique, the algorithm may be disengaged for an extended period; e.g., until shortly after all wheels are in contact with the runway.

The compensation algorithm should not be completely disengaged during vehicle rollout because high-gain tasks may exist during this period of operation. This was amply demonstrated by astronauts investigating the Space Shuttle's nose wheel steering system. Indeed, this aggressive piloting task is a good example of the need for a visual scene compensation algorithm.

At Ames we have not experienced other transient phenomena that excite the visual presentation during compensated operations. However, in a research environment it is a sure bet that they will occur in the future. Since the compensation algorithm may be turned off and on at leisure, a similar solution will probably exist.

Steady-State Noise

MILS-SPEC turbulence is generated in simulation models at Ames by filters which are driven by Gaussian noise. This noise extends over the entire

Nyquist bandwidth $F_{NYQ} = 1/2T$. The spectral form of these filters is such that (except for very small T) measurable power remains beyond the compensation algorithm's crossover frequency, especially in the angular gust velocities given by P_g, Q_g, and R_g. When high levels of turbulence are selected, some blurring of the visual presentation may be perceived.

A solution to the steady-state noise problem is to prefilter the high-frequency components, as shown in Fig. 1. The turbulence model is driven by a uniformly distributed, random noise source, where the distribution is shaped to be Gaussian. The prefilter operates upon this noise, so that phase is immaterial. Phase becomes important only when the outputs from the linear turbulence filters are used to create correlated angular disturbances.

Turbulence prefiltering is the inverse of the integration-plus-compensation operation, where high frequencies are attenuated prior to their eventual amplification by the CGI compensation algorithm. Within the mathematical model the high-frequency turbulence components remain attenuated (see Fig. 1).

In order to demonstrate the turbulence prefiltering operation, a worst-case set of parameters is used. This set consists of a vehicle velocity of 400 ft/sec, a scale length of 100 ft, and a span of 20 ft. Under these conditions of equal parameters for each of the turbulence dimensions, the gust velocities $V_g = W_g$, and to a close approximation, $R_g = Q_g$. Hence, only four of the six turbulence velocities are presented in Fig. 5. The three parametric curves are (1) the original MILS-SPEC continuum transfer functions, (2) the discrete realizations using zero-order data holds, and (3) the discrete realizations using the prefilters.

In Fig. 5 it is seen that the pass-band behavior remains close to that of the continuum model, whereas beginning with the crossover frequency, the prefilters produce a fourth order roll off.

TRANSPORT DELAY MEASUREMENT

Depending upon the particular CGI system and facility implementation, the value for transport delay may or may not be known to sufficient accuracy for use in the compensation algorithm. In this section a technique is described for the measurement of the delay. Measurement is by the pilot (experimenter), effectively including the complete simulation environment. The driving signals originate from position commands within the mainframe computer. Theoretical values for transport delay are computed, and these are compared to measured data.

Beam-Splitter Experiment

Transport delays between two originally concurrent signals are measured in the beam-splitter experiment. This is done by slowing down the faster of

the two systems by a measurable amount, where the faster system has a known delay, as discussed below. The "slower system" is a CGI image and the "faster system" is an analogous oscilloscope image.

• A CGI image is utilized in which a sharp line is observed well within the field of view. This sharp line is perhaps a monolith or the top of a building in the scene presentation. When measurements are taken, the observer's viewpoint (e.g., aircraft altitude) oscillates between two selected positions, always retaining the sharp line within a restricted field of view.

Superimposed on the CGI presentation via an optical beam splitter, is a line generated by an oscilloscope, the position (on the screen) of which is proportional to input voltage. This superimposed image is calibrated to be coincident with the CGI's "sharp line" at both extrema of travel. The facility configuration is as depicted in Fig. 6.

Events in the mainframe computer are given at discrete time points $t_k = kT$ ($k = 0, 1, 2, \ldots$). A sinusoidal signal Q_c is sent via a digital-to-digital (D/D) channel to the CGI system,

$$Q_c = \sin[\omega kT] \tag{17}$$

and a related sinusoidal signal Q_d is sent to the oscilloscope via a D/D channel followed by a digital-to-analog (DAC) conversion:

$$Q_d = \sin[\omega(kT - d)] \tag{18}$$

These real-time signals are transmitted at approximately the same time during each computer cycle; different origination times have to be considered when relative delays are computed.

An example experiment was performed to illustrate path differences. A particular CGI system called the "CT5A" was utilized. The particular facility configuration has a "communication computer" between the mainframe computer and the digital-to-analog conversion interface, as is shown in Fig. 6. This slightly complicates the determination of the oscilloscope's temporal path length.

Oscilloscope Path

Depending upon the value of "d" in equation (18), an observer will note phase differences in the dynamic positions of the two images caused by the different temporal path lengths of Q_c and Q_d. Fortunately, the temporal path length of the signal driving the oscilloscope is known within a small tolerance. This length is caused by digital-to-analog conversion (DAC) with a zero-order hold, given by $T_{DA} = T/2$, and by a facility overhead (Ref. 7) of approximately $T_{OH} = 6$ msec, caused by the communication computer. Since the oscilloscope's electronic delay is negligible, the total delay in creating the voltage-proportional oscilloscope line is T_{VP}

$$T_{VP} = T_{DA} + T_{OH} = \frac{T}{2} + 6 \quad \text{(msec)} \tag{19}$$

Figure 6 shows a manually adjustable potentiometer for use by the observer to alter the value of "d" in equation (18) until the two separate sinusoidal images appear temporally coincident. In practice this process becomes tedious for observers at frequencies both below and above a limited range. Over this limited range, which is much less than the "pass band" used in the algorithm, an observer synchronizes the displayed signals by adjusting the value of "d." Because the system manifests only pure transport delay, "d" is theoretically independent of the oscillation frequency.

CGI Path

A CGI system typically consists of a series of parallel processors, each running at the same rate. These processors are arranged in a "pipeline" configuration, so that for N processors, each operating at T_p, the temporal delay in processing data would be NT_p. However, this is modified by two other considerations. First, the last processor is a video generator, where every other line is first drawn and then the intermediate lines are filled in. It is thus perceived by an observer that the picture is created in roughly half the time required for video generation. Hence, the actual pipeline delay is given by $(N - 1/2)T_p$, as shown in Fig. 6.

The second consideration, discussed in detail in Ref. 1, is that the asynchronous operation of the mainframe and CGI computers produces a non-stationary delay T_{AS} that is a complex function of the ratio of sampling frequencies. If the CGI computer does not use a smoothing algorithm, the average value of this asynchronous delay may be used to approximate this delay source. For the experimental data given here the smoothing operation was disabled in order to avoid data contamination. Hence, the average asynchronous delay must be considered.

For normal simulation operations, however, a smoothing algorithm is implemented in the CGI computer to account for asynchronous data transfers; and in this case $T_{AS} = 0$ (approximately).

For purposes of demonstrating the average value for asynchronous delay, consider the first pipeline processor (scene management computer) to be operating at 50 Hz (a cycle time of 20 msec). Also, consider a mainframe computer cycling at $T = 36$ msec. The important ratio from Ref. 1 (which must be computed with its lowest common denominator) is given by the ratio of the mainframe sample rate to that of the first pipeline processor:

$$\frac{F}{F_P} = \frac{T_P}{T} = \frac{M}{N} = \frac{20}{36} = \frac{5}{9} \quad \text{(l.c.d.)} \tag{20}$$

Having found the value $N = 9$, the average asynchronous time delay may then be computed (Ref. 1):

$$T_{AS} = \frac{T}{2}\left(1 - \frac{1}{N}\right) = 16 \quad \text{(msec)} \tag{21}$$

This average value applies to the data given in Fig. 7 because the CGI's smoothing operation was disabled.

The 50 Hz rate is from the CT5A CGI system. Four processors are used in its pipeline, so that under the assumed conditions the theoretical delay for this system is approximately:

$$T_{CGI} = (4 - 1/2)T_P + T_{AS} = 86 \quad \text{(msec)} \tag{22}$$

Experimental Results

From the above discussion, the beam-splitter experiment is expected to produce a value of "d" that is approximately

$$d = T_{CGI} - T_{VP} = 86 - 24 = 62 \quad \text{(msec)} \tag{23}$$

Figure 7 displays raw data from a sample experiment. The average value for all 12 frequency points and 5 different observers is 57.2 msec, with a standard deviation of 6.2 msec.

Experiment thus compares well with the theoretical value of 62 msec, as given in equation (23). The transport delay T_{CGI} in equation (22) is thus verified. The small offset probably involves the assumption about "perceiving the picture in half the time," where a discussion of perception is beyond the scope of this paper. Also, the last processor (video generator) may not be the limiting factor in the CGI pipeline; i.e., the complete picture may be drawn in less than 20 msec. Finally, it is only an assumption that an observer will perceive the average value of T_{AS} when the CGI smoothing operation is not used.

Parameter Selection

In the above discussion, the averaged value $T_{CGI} = 86$ msec was theorized and shown to agree with experiment to a fair approximation. It is questionable, however, whether T_{CGI} is the proper value to implement in the compensation filter, i.e., "P" in equation (13). The proper value depends upon the desired point of synchronization for the visual display. If CGI smoothing is not used, and it is required that the display be synchronized with the mainframe outputs, then $P = 86$ msec as in equation (22).

However, if it is desired that the visual display be synchronized with the outputs from the digital-to-analog converters (DACs), then both the communication computer's delay and that which is due to DAC conversion should be subtracted in computing P; this would require that P = 62 msec.

The smoothing operation is generally enabled for real-time flight simulation. This means that the appropriate value is P = 70 msec for CGI synchronization with the mainframe, or P = 46 msec for synchronization with DAC outputs.

One additional factor arises in selecting the value of P for use in the compensation algorithm. Pilot commands are usually transmitted by analog-to-digital converters (ADCs), which introduce a time delay of T/2. The matrix of possible selections for the parameter P becomes large when this is considered.

The cycle time "T" is generally not selected from the considerations dealt with in this paper. From the relationships of equation (13), however, it should be appreciated that smaller sample times produce improved performance. This is usually true for discrete models, although exceptions do exist.

The third and final parameter in the compensation algorithm is the crossover frequency "F_o." From equation (13) and Fig. 4, reductions in F_o reduce the pass-band phase errors significantly. This is caused by the compound effect of decreasing both θ_o and ψ_o. Since a particular facility configuration (and "synchronization philosophy") will specify the value for P, and a particular mathematical model will specify the value for T, the maximum pass-band phase error then becomes a function of only F_o. Figures 2 and 4 may be used to determine the pertinent values. For large cycle times, for instance, the only alternative for good pass-band performance may be a reduction in F_o.

According to Ref. 4, researchers have been concerned with frequencies an order of magnitude smaller than those examined here. This means that considerable latitude may be taken with F_o. If reducing F_o is tolerable, the Ames Compensation Algorithm can compensate for very large CGI transport delays.

Figures 8, 9, and 10 are provided as illustrations of the parametric selection process. In Fig. 8, the performance is given when compensation is attempted for a CGI delay P of 100 msec using a selected crossover frequency of 3 Hz. For cycle times above about 30 msec, the maximum pass-band phase error may not be tolerable. One alternative, the reduction of the compensation interval, is presented in Fig. 9. For the complete range of cycle times in this figure the pass-band phase error is negligible. However, if the prediction interval cannot be reduced from the 100 msec value (and the cycle time cannot be reduced), then the proper alternative is the reduction of F_o. This is illustrated in Fig. 10, where a slight reduction in crossover frequency severely decreases all pass band errors.

CONCLUSIONS

Performance within a restricted frequency bandwidth may be improved at the expense of performance beyond the bandwidth. When activity beyond the bandwidth is attenuated because of other considerations, this trade off may be used in the development of a compensation algorithm, such as that described here.

The algorithm delivers high-fidelity, compensated CGI drive signals over the human-factors bandwidth, and can dramatically improve pilot control for high-gain tasks such as precision hovering and stationkeeping.

REFERENCES

(1) R. E. McFarland, "CGI Delay Compensation," NASA TM 86703, Jan. 1986.

(2) D. R. Rolston, "Sinusoidal Integration for Simulation of Second Order Systems," AIAA Flight Simulation Technologies Conference Collection of Technical Papers, Niagara Falls, NY, June 1983, pp. 52-63.

(3) D. R. Gum and E. A. Martin, "The Flight Simulator Time Delay Problem," AIAA Paper No. 87-2369-CP, AIAA Simulation Technology Conference, August 17-19, 1987, Monterey, CA, p. 1.

(4) G. B. Churchill and R. M. Gerdes, "Advanced AFCS Developments on the XV-15 Tilt Rotor Research Aircraft," 40th Annual Forum of the American Helicopter Society, Arlington, VA, Paper No. A-84-40-10-4000, May 16-18, 1984, p. 10.

(5) W. F. Jewell, W. F. Clement, and J. R. Hogue, "Frequency Response Identification of a Computer-Generated Image Visual Simulator with and without a Delay Compensation Scheme," AIAA Paper No. 87-2435, AIAA Flight Simulation Technologies Conference, Aug. 17-19, 1987, Monterey, CA, p. 73.

(6) R. E. McFarland, "Anticipation of the Landing Shock Phenomenon in Flight Simulation," NASA TM 89465, Sept. 1987.

(7) W. B. Cleveland, "A Time Lag Study of the Vertical Motion Simulation Computer System," NASA TM 81306, Aug. 1981, p. 12.

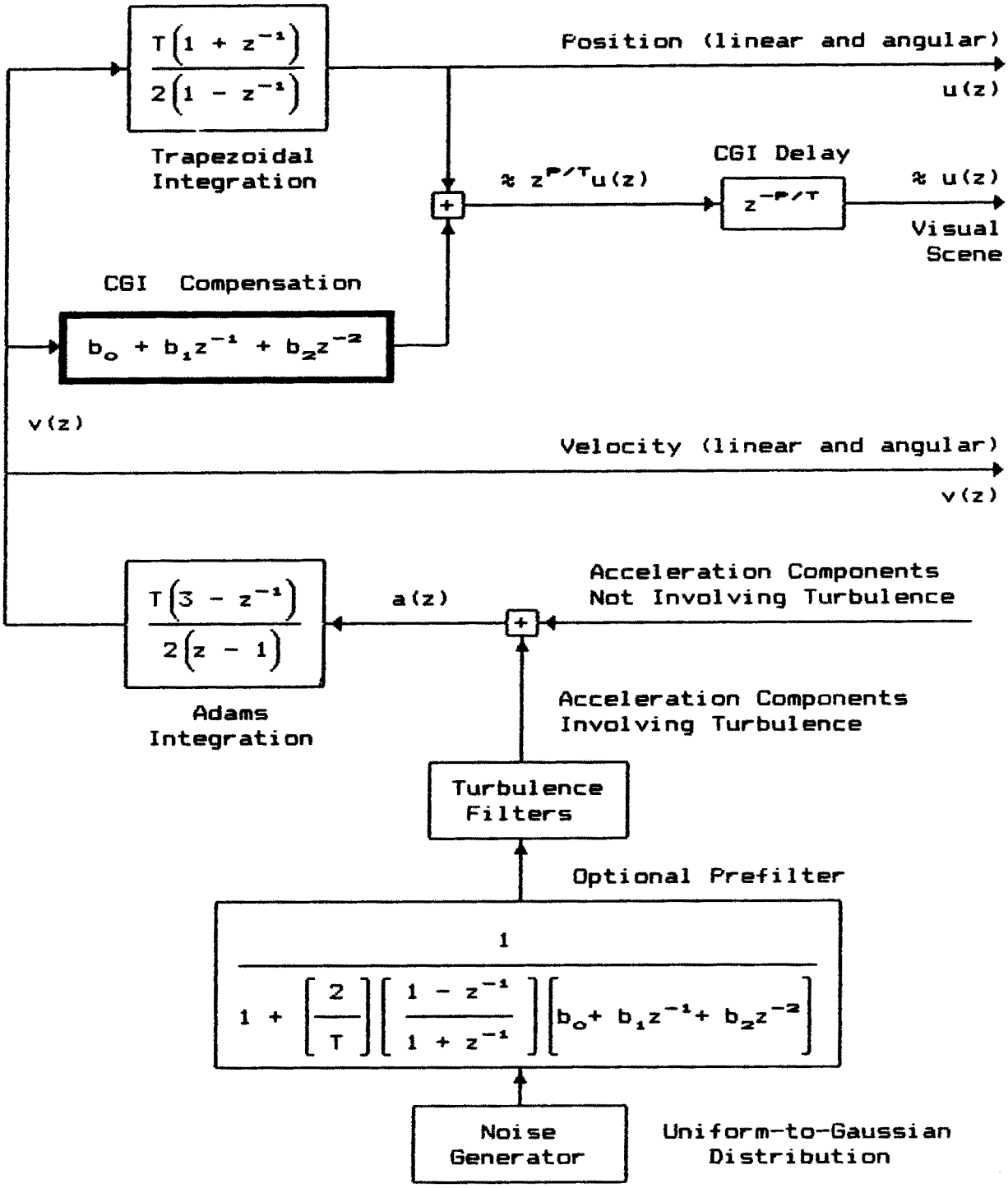

Fig. 1 Ames state integration scheme, compensation algorithm and optional turbulence prefilter

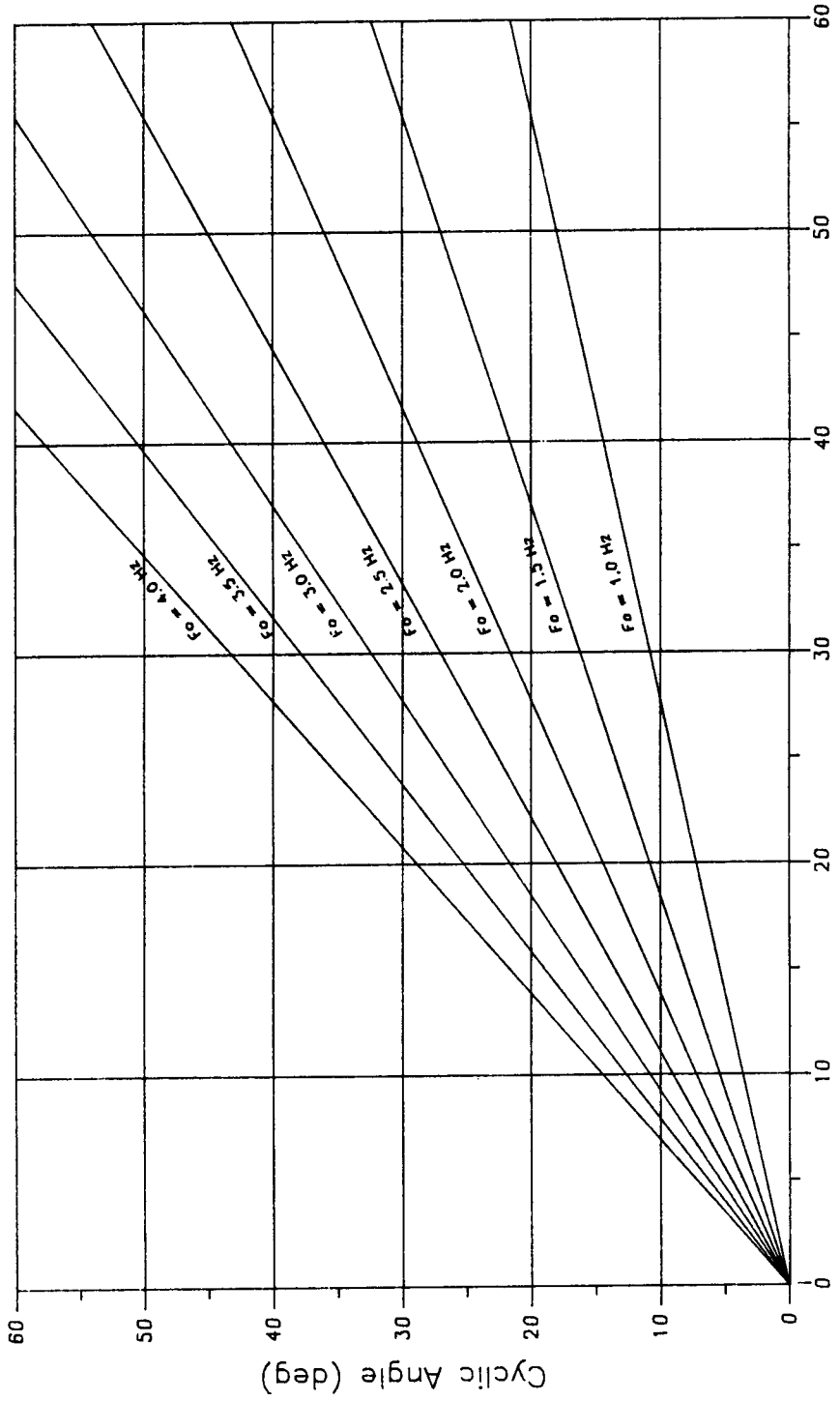

Fig. 2 Cyclic angle vs cycle time at various crossover frequencies

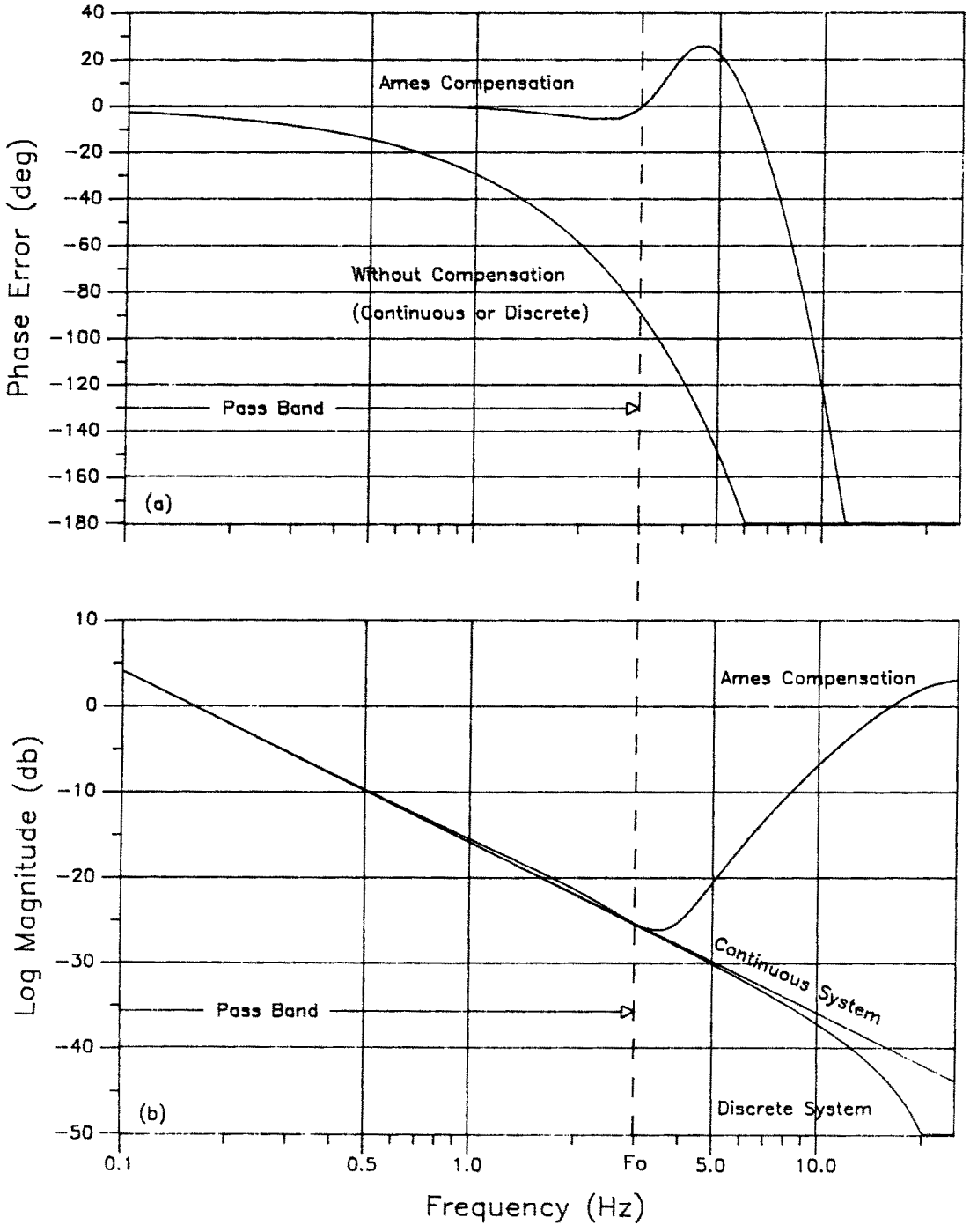

Fig. 3 Compensation performance, T = 20 msec, P = 83 msec, F_o = 3 Hz. (a) Phase error, (b) magnitude

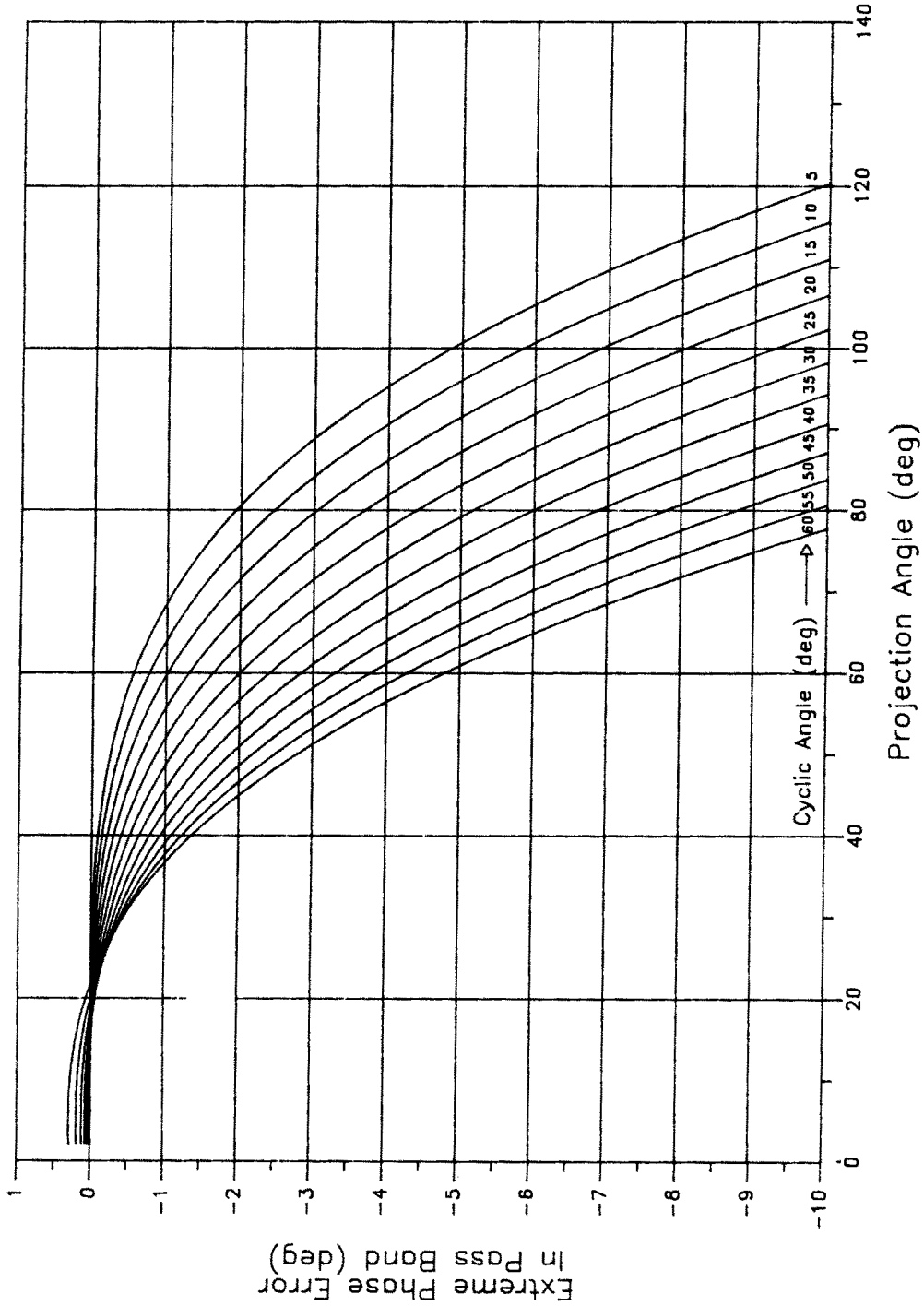

Fig. 4 Worst-case phase error in pass band

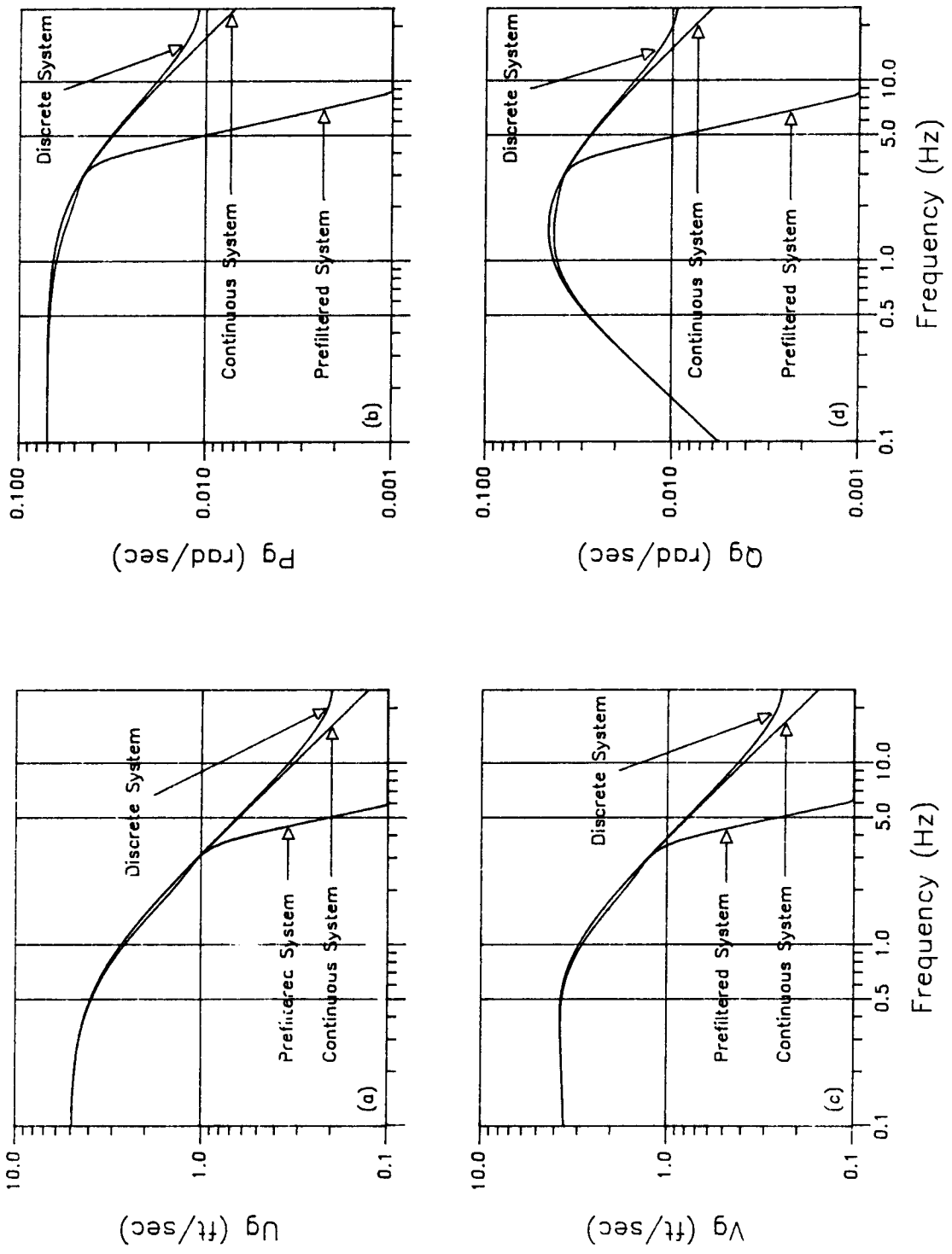

Fig. 5 Turbulence prefilter characteristics, T = 20 msec, P = 83 msec, F_o = 3 Hz. (a) Ug, (b) Pg, (c) Vg, (d) Qg

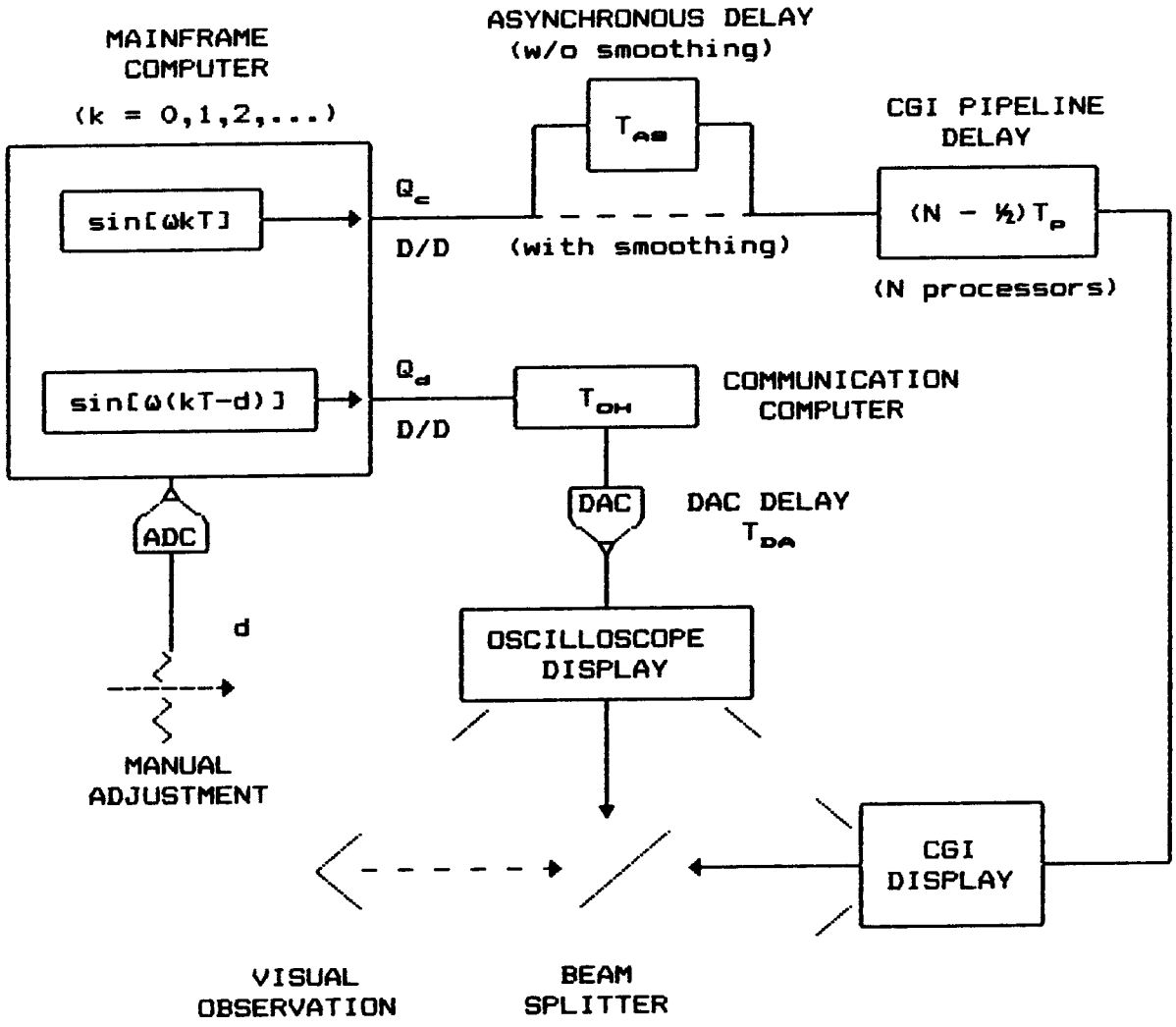

Fig. 6 Beam-splitter experiment

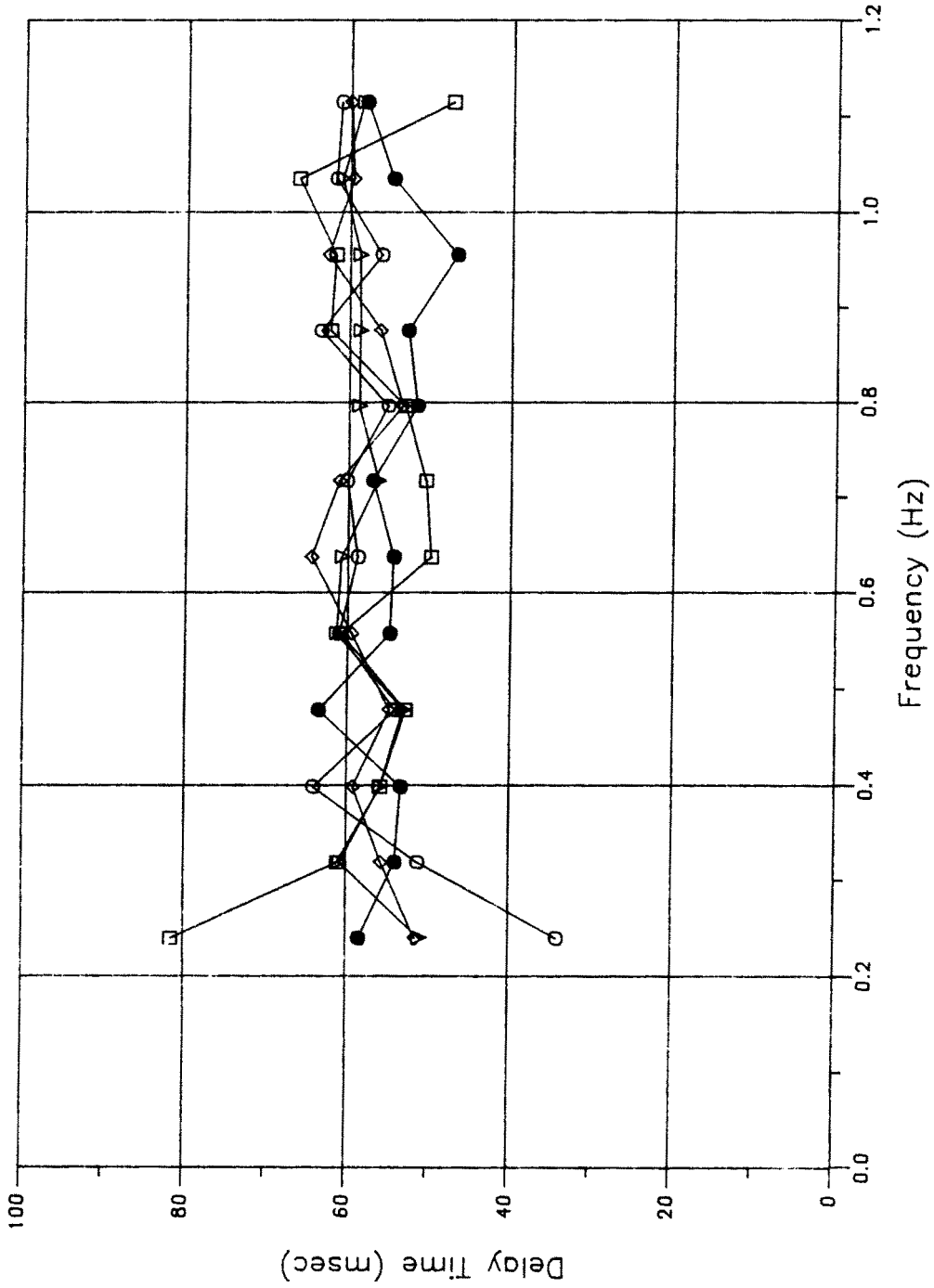

Fig. 7 Beam-splitter measurements

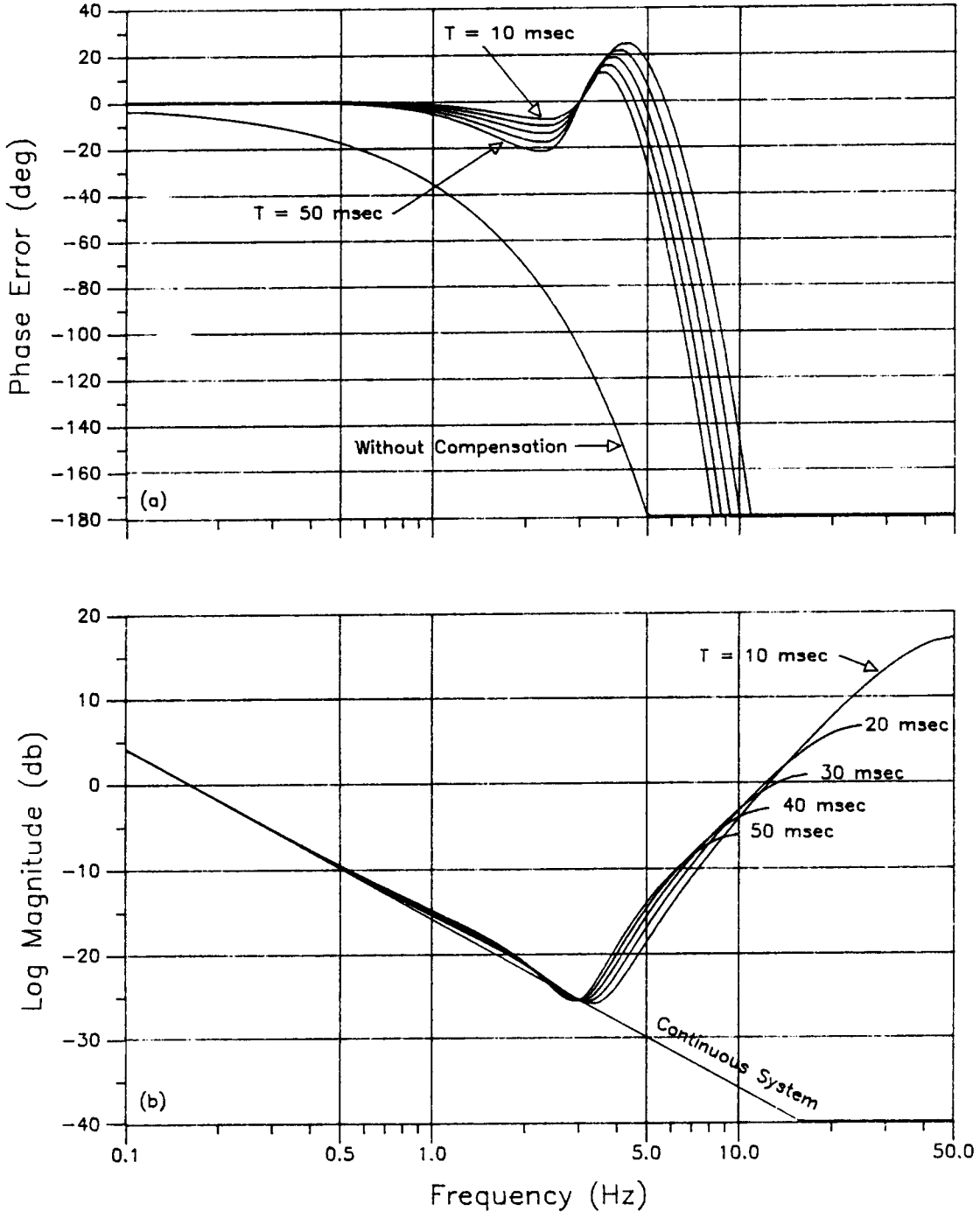

Fig. 8 100 msec prediction, F_o = 3 Hz. (a) Phase error, (b) magnitude

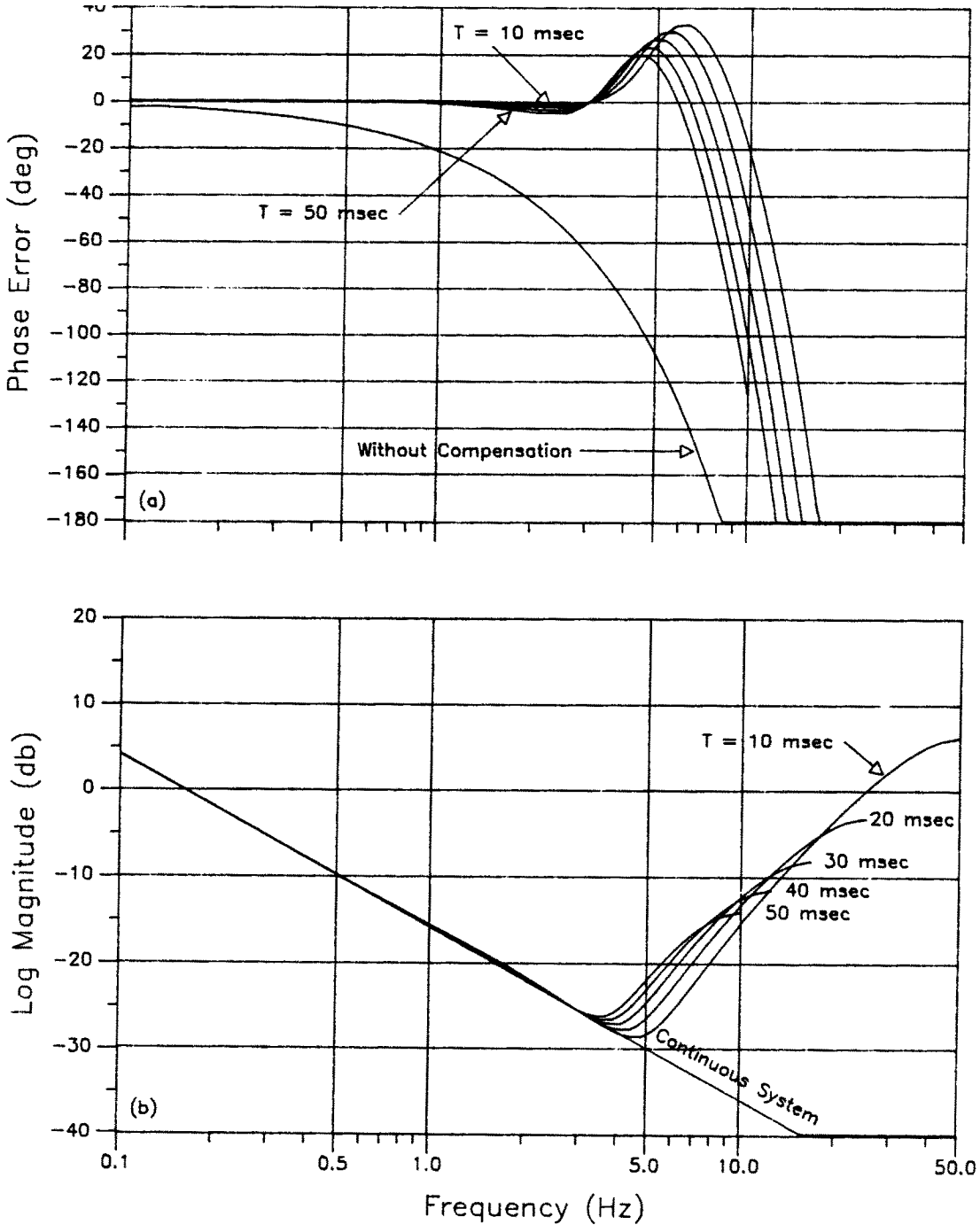

Fig. 9 60 msec prediction, F_o = 3 Hz. (a) Phase error, (b) magnitude

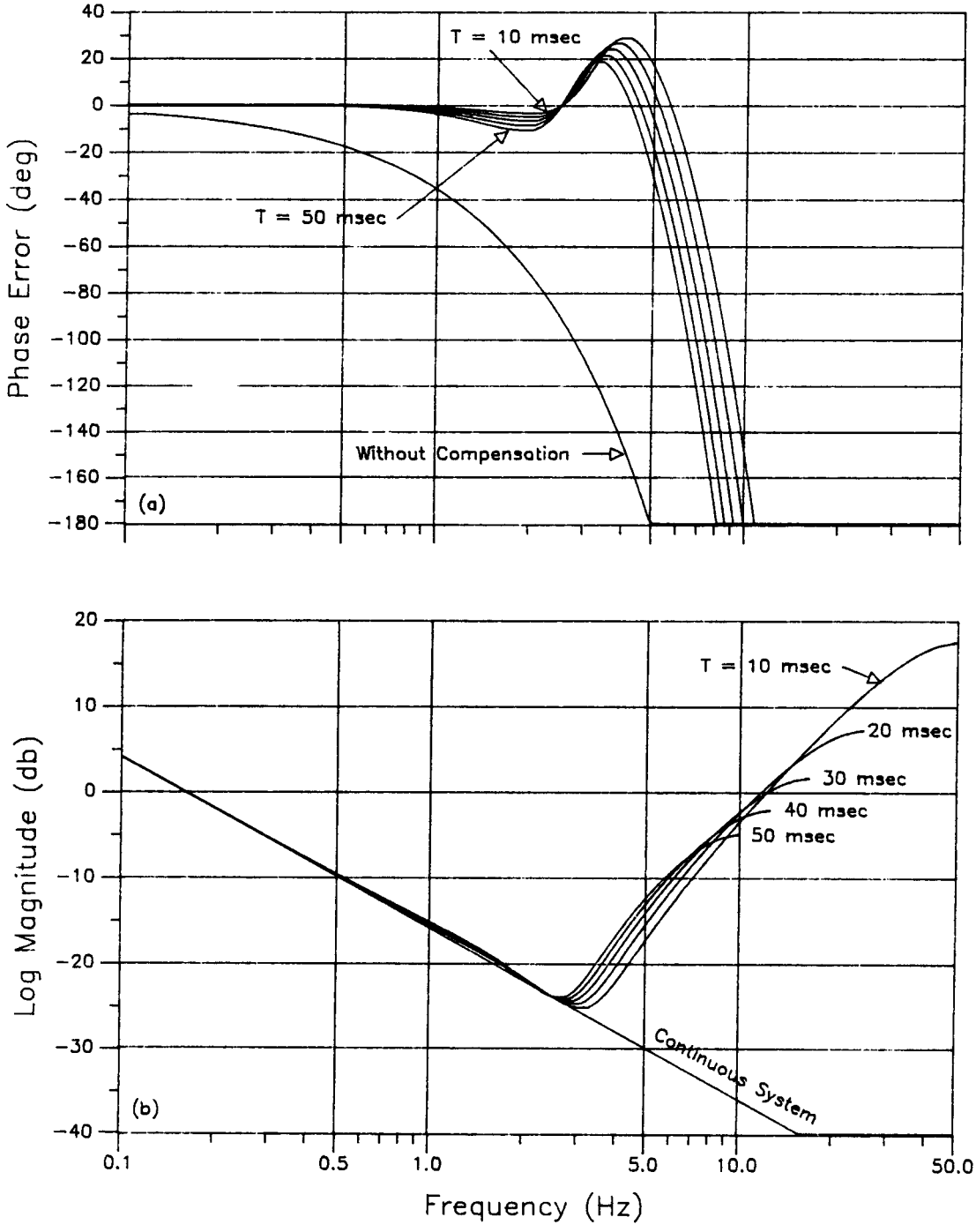

Fig. 10 100 msec prediction, F_o = 2.5 Hz. (a) Phase error, (b) magnitude

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM 100084		

4. Title and Subtitle	5. Report Date
Transport Delay Compensation for Computer-Generated Imagery Systems	January 1988
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Richard E. McFarland	A-87385
	10. Work Unit No.
	505-67-51

9. Performing Organization Name and Address	11. Contract or Grant No.
Ames Research Center Moffett Field, CA 94035	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, D.C. 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

Point of contact: Richard E. McFarland, Ames Research Center, MS-243-5, Moffett Field, CA 94035 (415) 694-6171 or FTS 464-6171

16. Abstract

In the problem of pure transport delay in a low-pass system, a trade-off exists with respect to performance within and beyond a frequency bandwidth. When activity beyond the band is attenuated because of other considerations, this trade-off may be used to improve the performance within the band. Specifically, transport delay in computer-generated imagery systems is reduced to a manageable problem by recognizing frequency limits in vehicle activity and manual-control capacity. Based upon these limits, a compensation algorithm has been developed for use in aircraft simulation at NASA Ames Research Center. For direct measurement of transport delays, a beam-splitter experiment is presented that accounts for the complete flight simulation environment. Values determined by this experiment are appropriate for use in the compensation algorithm. The algorithm extends the bandwidth of high-frequency flight simulation to well beyond that of normal pilot inputs. Within this bandwidth, the visual scene presentation manifests negligible gain distortion and phase lag. After a year of utilization, two minor exceptions to universal simulation applicability have been identified and subsequently resolved.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Transport delay Compensation CGI systems Extrapolation High-gain tasks	Unclassified - Unlimited Subject category: 05

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	28	A02